VideoGEM: Training-free Action Grounding in Videos

Felix Vogel[1], Walid Bousselham[1,2], Anna Kukleva[3], Nina Shvetsova[1,2,3], Hilde Kuehne[1,2,4]
[1] Goethe University Frankfurt, [2] Tuebingen AI Center/University of Tuebingen, [3] MPI for Informatics, SIC, [4] MIT-IBM Watson AI Lab

Abstract

Vision-language foundation models have shown impressive capabilities across various zero-shot tasks, including training-free localization and grounding, primarily focusing on localizing objects in images. However, leveraging those capabilities to localize actions and events in videos is challenging, as actions have less physical outline and are usually described by higher-level concepts. In this work, we propose VideoGEM, the first training-free spatial action grounding method based on pretrained image- and video-language backbones. Namely, we adapt the self-self attention formulation of GEM [2] to spatial activity grounding. We observe that high-level semantic concepts, such as actions, usually emerge in the higher layers of the image- and video-language models. We, therefore, propose a layer weighting in the self-attention path to prioritize higher layers. Additionally, we introduce a dynamic weighting method to automatically tune layer weights to capture each layer's relevance to a specific prompt. Finally, we introduce a prompt decomposition, processing action, verb, and object prompts separately, resulting in a better spatial localization of actions. We evaluate the proposed approach on three image- and video-language backbones, CLIP, OpenCLIP, and ViCLIP, and on four video grounding datasets, V-HICO, DALY, YouCook-Interactions, and GroundingYouTube, showing that the proposed training-free approach is able to outperform current trained state-of-the-art approaches for spatial video grounding. [1]

1. Introduction

Spatial localization of actions in videos has been a longstanding topic in video analysis [6, 10, 19, 23, 27, 28, 30, 34, 35]. While early approaches were trained with human-annotated bounding boxes and mainly focused on detecting humans and classifying respective actions [8, 26], recent approaches [3, 29, 33] are trying to leverage the localization properties of vision-language models without the need of bounding box annotations, learning localization only from image- text or video-text pairs. However, the localization of actions based on such weak supervision presents some significant challenges: unlike objects, which usually have a clear, unambiguous outline, actions often lack such region properties and exhibit diverse, context-dependent semantics. Additionally, actions often refer to interactions between entities and can comprise multiple objects or people over different timespans as shown in Figure 1. This variability makes it hard to capture the various visual representations of an action, especially based on web-crawled image- or video-text pairs. As a result, spatial video grounding methods [3, 24, 29] still need to be specifically trained, e.g. with a localization loss, to perform spatial localization.

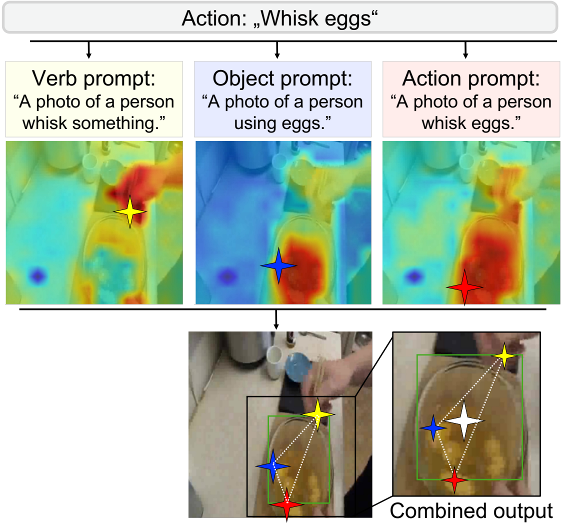
Action: „Whisk eggs“
Verb prompt:
“A photo of a person whisk something.”
Object prompt:
“A photo of a person using eggs.”
Action prompt:
“A photo of a person whisk eggs.”
Combined output

Figure 1. **Prompt decomposition and combination.** First, we decompose the action query into verb, object, and action prompts. For each component, we then predict locations corresponding to the highest values on the heatmaps (red-high, blue-low). To determine the final location (white star), we calculate the center point of these individual predictions. The red, blue, and yellow stars represent the predicted locations for the action, object, and verb prompts, respectively, while the dark green bounding box represents the annotated ground truth.

Compared to that, another line of work focuses on training-free localization of semantic concepts in images [2, 15, 40] by proposing various training-free adaptations of vision-language models to perform object localization.

[1] Code is available at https://github.com/felixVogel02/VideoGEM

Namely, CLIPSurgery [15] proposes an alternative pathway based on value-value attention which was extended by Grounding Everything Module (GEM) [2] to a self-self attention pathway. While those methods work well for object localization in image data, action localization requires models to capture contextual cues beyond object boundaries.

To address this problem, we introduce VideoGEM, the first training-free method for spatial activity grounding in videos. Inspired by GEM [2], which focuses on spatial object localization, VideoGEM extends this approach to spatially localize activities within video content. First, we extend the self-self attention pipeline to process multiple frames for action localization on video data for applying video backbones. In this setup, self-self attention automatically spans multiple frames, aggregating attention across both space and time. However, pretrained image-language but also video-language backbones struggle to capture abstract concepts such as actions. To address this problem, we analyze the self-self attention pipeline of several backbones, showing that abstract concepts such as actions and verbs usually arise in higher layers. We, therefore, propose to weight layers of the self-self attention pipeline based on a mixture of static and dynamic weights: while static weights give more weight to higher layers, dynamic weights are adjusted based on the relevance of a layer for each prompt.

We further observe that vision-language models often show a strong object bias when they are prompted with verb-object combinations [2, 31, 38]. To counteract this and focus the model on both verb and object, we propose a prompt decomposition. To this end, the *verb* and *object* of the *action* description are extracted and separately prompted in addition to the original action. We then compute the center points of the resulting individual predictions for *verb*, *object*, and *action*, and consider the weighted mean of all three center point predictions (see Figure 1). This accounts for the fact that verbs might focus more on hands, while object heatmaps automatically capture the object.

We evaluate our approach on three pretrained backbones, CLIP [21], OpenCLIP [22], and ViCLIP [32, 36] and on four action grounding datasets, V-HICO [14], Daly [35], YouCook-Interactions [29], and GroundingYouTube [3]. It shows that VideoGEM allows image and video backbones to spatially ground actions in a training-free zero-shot manner and that they are able to outperform models specially trained for this setup. We further analyze the impact of the proposed components, including the effect of video vs image processing for self-self attention as well as the impact of layer weighting and prompt decomposition, showing how those factors contribute to the final performance.

We summarize our contributions as follows: (1) We propose VideoGEM, the first training-free method for spatial action grounding in videos that adapts self-self attention to the video domain. (2) To capture higher-level sematic concepts such as actions, we propose a mixture of static and dynamic weights that prioritizes layers according to their relevance for capturing such concepts. (3) We propose prompt decomposition to address the object bias in vision-language models, allowing self-self attention to consider actions, verbs, and objects independently. (4) We show that VideoGEM outperforms even fine-tuned localization methods and provide extensive analysis of all the components.

2. Related Work

Spatial Video Grounding. Spatially localizing actions in videos without the need to explicitly label respective instances has drawn significant attention in the last years. As one of the first works, CoMMA (Contrastive Multilayer Multimodal Attention) [29] proposes a multilayer cross-modal attention network that obtains an attention heatmap via attention rollout, where the predicted location is the maximum attention. Moreover, the authors propose the YouCook-Instructions dataset, based on YouCook2 [41], to evaluate spatial grounding for models pretrained on YouTube cooking data. Compared to that, TubeDETR [37] uses a space-time decoder, decoding the video-text features into a spatiotemporal object tube which includes box annotations per frame. STCAT [9] proposes a Spatio-Temporal Consistency-Aware Transformer that also uses a vision and text encoder, followed by cross-model interaction. To create the object tube, the authors generate multi-modal templates to guide the decoder, followed by a prediction head. Recently, VideoGrounding-DINO [33] extends GroundingDINO [16] to videos. Finally, What-When-Where [3] proposes a global and local loss together with a frame selection mechanism to detect actions in untrimmed videos in space and time. The authors also evaluate on a new benchmark, GroundingYouTube, which is based on MiningYouTube [11], to evaluate spatiotemporal grounding in untrimmed videos, and also use the annotated segments to evaluate spatial grounding alone. In contrast, our method is training-free, using the original backbone weights without additional fine-tuning, whereas the previous methods rely on additional training of either a new projection head or the fine-tuning of the full model.

Training-free Vision-Language Grounding. The success of large-scale vision-language models like CLIP has generated significant interest in applying them to tasks such as open-vocabulary object localization. In this context, a special line of methods focuses on training-free localization [2, 15, 40], thus adapting pretrained vision-language models to handle localization tasks without changing the weights of the pretrained model. MaskCLIP [40] achieves this by removing the Multi-Layer Perceptron (MLP) in the vision transformer's final layer, using the last value projection to capture dense patch-level features. Building on this, CLIPSurgery [15] introduces a parallel "surgery pathway" alongside the standard CLIP vision transformer (ViT) backbone, which operates with value-value attention instead of the usual query-key attention and connects outputs from multiple layers through residual connections. Consistent with MaskCLIP [40], CLIPSurgery omits the final MLP, directly applying value-value attention. GEM [2] further improves the concept of value-value attention by generalizing it to self-self attention, not only using value-value attention but also query-query and key-key attention together with a set of regularizations. We extend the GEM model and the self-self attention mechanism by adapting it to video input and introducing a weighting and a prompt decomposition mechanism, applying it to action localization.

3. Method

In the following, we first review the GEM self-self attention pipeline in Section 3.1. We then discuss its extension to videos in Section 3.2, the proposed layer weighting in Section 3.3, and the prompt decomposition in Section 3.4.

3.1. Background (GEM)

In the original ViT paper [5], the attention operation is computed as follows:

$$\begin{aligned} \mathbf{A} &= \mathrm{softmax}\left(\frac{\mathbf{X}\mathbf{W}_q(\mathbf{X}\mathbf{W}_k)^T}{\tau}\right), \\ \mathbf{O} &= \mathbf{A}\cdot(\mathbf{X}\cdot\mathbf{W}_v), \end{aligned} \tag{1}$$

where $\mathbf{X} = (x_i)_{i\leq N} \in \mathbb{R}^{N\times d}$ represents the patch tokens output from the ViT, N is the number of visual tokens, d is the dimension of each token and $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v \in R^{d\times(h*d_h)}$ are the projection matrices, where h is the number of heads in the attention and d_h is the dimension of each head. The Grounding Everything Module (GEM) [2] introduces a parallel pathway that operates alongside the original trained ViT while sharing its weights. This pathway replaces the standard self-attention mechanism with a self-self attention operation defined as follows:

$$\mathbf{A}_{ss} = \mathrm{softmax}\left(\frac{(\mathbf{X}\mathbf{W}_{\text{proj}})(\mathbf{X}\mathbf{W}_{\text{proj}})^\top}{\tau}\right), \tag{2}$$

where $\mathbf{W}_{proj} \in \{\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v\}$, and τ is the temperature. The self-self attention is applied iteratively J times on L^2 normalized visual tokens. For input visual tokens $\mathbf{X} \in \mathbb{R}^{N\times d}$, we denote $\mathbf{P}^{(j)}$ as the output of the self-self attention at iteration j:

$$\begin{cases} \mathbf{P}^{(0)} = \dfrac{\mathbf{X}\mathbf{W}_{proj}}{\|\mathbf{X}\mathbf{W}_{proj}\|_2}, \\ \tilde{\mathbf{P}}^{(j)} = \mathrm{softmax}\left(\dfrac{\mathbf{P}^{(j-1)}(\mathbf{P}^{(j-1)})^T}{\tau}\right)\mathbf{P}^{(j-1)}, \\ \mathbf{P}^{(j)} = \dfrac{\tilde{\mathbf{P}}^{(j)}}{\|\tilde{\mathbf{P}}^{(j)}\|_2}. \end{cases} \tag{3}$$

The final output $\mathbf{O}_{ss}$ is obtained by applying the attention matrix to the values:

$$\mathbf{O}_{ss} = \mathrm{softmax}\left(\frac{\mathbf{P}^{(J)}\cdot(\mathbf{P}^{(J)})^T}{\tau}\right)\cdot V. \tag{4}$$

The final self-self attention output is obtained by averaging the query-query, key-key, and value-value attention:

$$\mathbf{O}_{qkv} = \frac{(\mathbf{O}_{qq} + \mathbf{O}_{kk} + \mathbf{O}_{vv})}{3}. \tag{5}$$

This formulation enables the patch tokens alignment with the Vision-Language Model (VLM) text encoder. Localization heatmaps can then be constructed by computing the cosine similarity between the text embedding of a prompt and the corresponding patch token representations.

3.2. GEM for Videos

We extend GEM to handle video inputs by adapting it to video-language models, using the ViCLIP backbone as an example. Given a video input, ViCLIP processes a sequence of $T = 8$ frames $\mathbf{F} = \{f_1, ..., f_T\}$. Each frame f_i is divided into N patches, resulting in a total of $T \times N$ patch tokens. These tokens are processed jointly across all frames through the transformer layers, allowing the model to capture both spatial and temporal relationships.

For a given input sequence, the patch tokens from all frames are concatenated into a single sequence $\mathbf{X} \in \mathbb{R}^{(T*N)\times d}$, where d is the embedding dimension. The self-self attention mechanism is then applied to this combined sequence of tokens. After that, we compute the cosine similarity between each patch token and the prompt embedding of the text encoder, resulting in a similarity score matrix $\mathbf{S} \in \mathbb{R}^{(T*N)}$. To generate the final localization output, we focus on a target frame f_t positioned such that we maintain temporal context from both past and future frames. The similarity scores for frame f_t are reshaped and interpolated to match the spatial dimensions of the input frame and min-max normalized to produce the final attention heatmap. Note, that the temporal information of the video is solely considered by the video backbone.

3.3. Layer Weighting

In the original GEM formulation, the self-self attention outputs are combined through residual connections, effectively

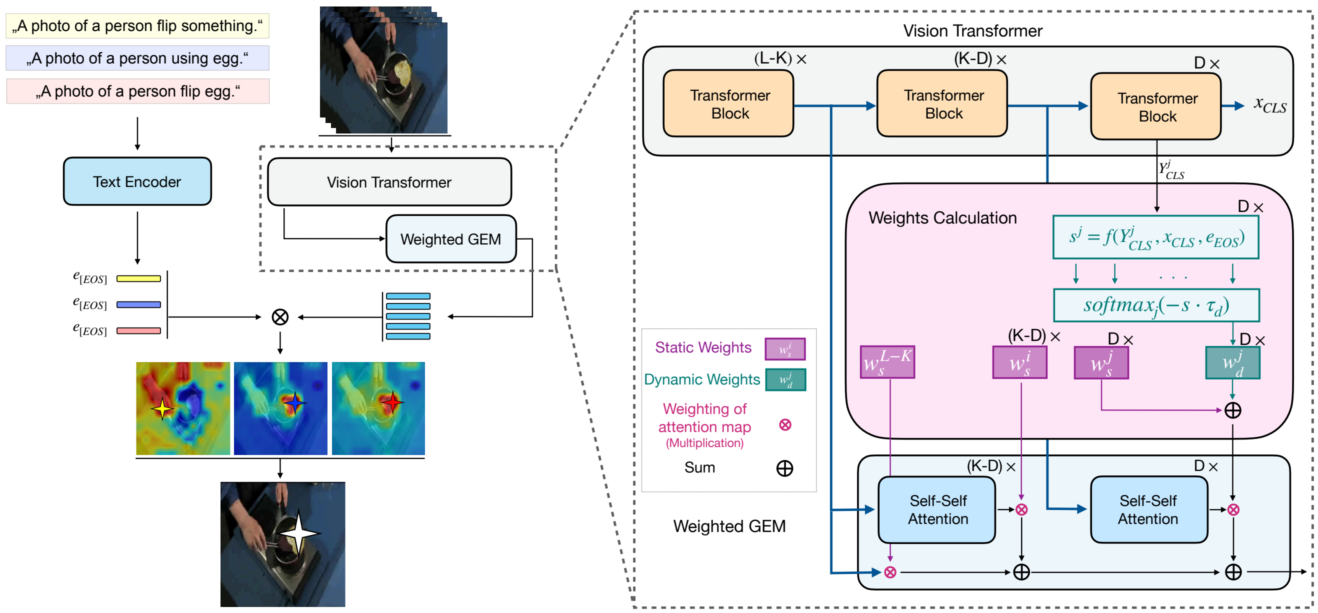

Figure 2. **Left: VideoGEM pipeline.** VideoGEM takes a video and its corresponding narration as input. Our *Weighted GEM* processes the input video alongside the vision transformer to generate the representative patch tokens. Decomposition of the input narration into verb prompt, object prompt, and action prompt (see Section 3.4 for details) are passed through the text encoder to obtain three *[EOS]* tokens, respectively. Then, three heatmaps are calculated as a similarity between patch tokens and the respective *[EOS]* tokens. We then aggregate the heatmaps into one final prediction by centering the individual predicted locations. **Right: Layer weighting.** In our *Weighted GEM* architecture, we apply a combination of static and dynamic weights. Dynamic weights are applied to the last D layers, while static weights are applied to the last K layers, with $K > D$. Additionally, the attention map X^{L-K} is weighted by a corresponding static weight w_s^{L-K}. All weighted outputs from the self-attention blocks are then summed with the weighted X^{L-K} attention map to produce representative patch tokens. The output patch tokens of weighted GEM are used for similarity calculation with the text, resulting in an attention heatmap.

giving equal importance to all layers in the final prediction. However, we observe that higher-level concepts, particularly actions, predominantly emerge in the higher layers of the network. Based on this, we propose a combination of static and dynamic weights to prioritize the contributions of different layers. We extend the definition of $\mathbf{X}$ from the previous section, to $\mathbf{X}^l \in \mathbb{R}^{(T*N)\times d}$ which represents the l-th transformer block output, where $l \in \{1, ..., L\}$ and L is the total number of transformer blocks. Let $\mathbf{Y}^l \in \mathbb{R}^{(T*N)\times d}$ denote the output of the l-th transformer block before the residual connection. Similarly, let $\mathbf{Z}^l \in \mathbb{R}^{(T*N)\times d}$ represent the output of the parallel self-self attention pathway at layer l before the residual connection. Then, when applied to the last K layers of the ViT, the output of GEM can be reformulated as follows:

$$\mathbf{O}_{GEM} = \mathbf{X}^{L-K} + \sum_{l=L-K+1}^{L} \mathbf{Z}^l. \tag{6}$$

Static Weighting. We first introduce static weights to assign specific importance to individual layers. For layer l, the static weight is defined as w_s^l for $l \in \{L-K, ..., L\}$ resulting in the new output:

$$\mathbf{O}_{stat} = w_s^{L-K} \cdot \mathbf{X}^{L-K} + \sum_{l=L-K+1}^{L} w_s^l \cdot \mathbf{Z}^l. \tag{7}$$

Practically, w_s^l increases monotonically with l.

Dynamic Weighting. We further introduce dynamic weights that adapt to the semantic requirements of each input prompt by analyzing each layer's contribution to the model's understanding of the prompted concept.

Let $\mathbf{x}_{CLS} = \mathbf{X}_0^L \in \mathbb{R}^d$ denote the [CLS] token representation from the final layer, which can be decomposed into the sum of residuals from all layers: $\mathbf{x}_{CLS} = \sum_{l=1}^{L} \mathbf{Y}_0^l = \sum_{l=1}^{L} \mathbf{Y}_{CLS}^l$. The text embedding of the prompt obtained from the text encoder is defined as $\mathbf{e}_{EOS} \in \mathbb{R}^d$. We measure each layer's importance by evaluating how its removal affects the alignment between visual and textual representations, for details see Section 4.4. Let $s = (s^{L-D+1}, ..., s^L)$ be the similarity vector where we compute the similarity score for layer l as:

$$s^l = \frac{(\mathbf{x}_{CLS} - \mathbf{Y}_{CLS}^l) \cdot \mathbf{e}_{EOS}}{\|\mathbf{x}_{CLS} - \mathbf{Y}_{CLS}^l\|_2 \cdot \|\mathbf{e}_{EOS}\|_2}, \tag{8}$$

The dynamic weights for the last D layers with $D \leq K$ are computed using the similarity vector s through a softmax operation with temperature τ_d. We define the dynamic weight for layer $l \in \{L-D+1, ..., L\}$ as:

$$w_d^l = \text{softmax}_l(-s \cdot \tau_d), \tag{9}$$

then the final output for dynamic weights is computed as:

$$\mathbf{O}_{dyn} = \mathbf{X}^{L-K} + \sum_{l=L-K+1}^{L-D} \mathbf{Z}^l + \sum_{l=L-D+1}^{L} w_d^l \cdot \mathbf{Z}^l. \quad (10)$$

It is important to note that the [CLS] token representations used for computing similarities are extracted from the original self-attention pathway rather than the self-self attention. It is motivated by the fact that the [CLS] token in the original model was specifically trained to align with the text encoder's [EOS] token representations during pre-training.

Combining Static and Dynamic Weights. To leverage the complementary benefits of both weighting schemes, we propose a unified approach that combines static and dynamic weights (see Figure 2). For a network with L layers, we apply static weights to the last K self-self attention layers and the previous self attention input while computing dynamic weights for the last D layers (where $D \leq K$). The combined weights w_c^l for $l \in \{L-K, ..., L\}$ are defined as:

$$w_c^l = \begin{cases} w_s^l - \dfrac{1}{D} + w_d^l & \text{if } l > L - D \\ w_s^l & \text{otherwise} \end{cases} \quad (11)$$

where w_s^l denotes static weight and w_d^l represents the dynamic weight as defined in Equation (9). We subtract $\frac{1}{D}$ from static weights when dynamic weights are applied to not increase the sum of weights. The final output incorporating these combined weights is computed as:

$$\mathbf{O}_{comb} = w_c^{L-K} \cdot \mathbf{X}^{L-K} + \sum_{l=L-K+1}^{L} w_c^l \cdot \mathbf{Z}^l \quad (12)$$

This weighted combination allows the model to adaptively balance the static prior knowledge about layer importance with dynamic, prompt-specific adjustments, particularly in higher layers where semantic concepts emerge.

3.4. Prompt Decomposition for Action Grounding

Action descriptions typically consist of two key components: A verb describing the action itself and one or more objects involved in the action. To effectively leverage this inherent structure, we propose a prompt decomposition method that processes these components separately while maintaining the context of the complete action description. Namely, we decompose each action query into three distinct components: a verb prompt, an object prompt, and the original action prompt. For consistent processing, we employ the following template format:

Prompt Templates

- Verb: *A photo of a person* $\{$*verb*$\}$ *something.*
- Object: *A photo of a person using* $\{$*object*$\}$.
- Action: *A photo of a person* $\{$*action prompt*$\}$.

In case a component, verb or object, is missing or cannot be extracted, we use fallback templates, *”A photo of a person doing something.”* for missing verbs and *”A photo of a person.”* for missing objects. For each prompt, we compute the similarity between its text embedding and the visual tokens, identifying the regions of highest activation. The model then determines a center point for each attention map, corresponding to the location with maximum similarity between the text and visual representations.

To combine these separate predictions into a final localization output, we employ a weighted averaging scheme that prioritizes the action prompt while incorporating information from the component-specific predictions (see Figures 1 and 2). Let $c_{verb}, c_{obj}, c_{act} \in \mathcal{R}^2$ denote the predicted center coordinates for the verb, object, and action prompts respectively, and $w_{verb}, w_{obj}, w_{act} \in \mathcal{R}$ their corresponding weights, where we assign higher weights to action prompts. The final prediction c_{dec} is computed as:

$$c_{dec} = w_{verb} \cdot c_{verb} + w_{obj} \cdot c_{obj} + w_{act} \cdot c_{act} \quad (13)$$

The weighted combination serves two purposes. First, it maintains the primacy of the action prompt's prediction while allowing refinement based on component-specific localizations. Second, if any single component prediction deviates from the true action center, the ensemble nature of the prediction helps maintain accuracy through the influence of the other components. This is particularly valuable for complex actions where the verb and object locations might provide complementary spatial information.

Extracting Verbs and Objects. For datasets with a label structure of *”verb_object”* verb and object can be directly retrieved. For natural language annotations as in YouCook-Interactions, we select the verbs and objects as the ones that are most likely visible in the input. To this end, we extract all verbs and objects from the action description with a natural language processing (NLP) tool. We then generate an individual prompt for each verb and object and apply the respective vision-language backbone to determine which verb and object show the highest similarity to the input.

4. Results

4.1. Datasets

We evaluate four video datasets for action grounding. Since our proposed method is training-free, we only consider the test set of the datasets for our evaluation, if there is one.

V-HICO (Videos of Humans Interacting with Common Objects) [14] has a test set containing 608 videos with annotated bounding boxes for a human performing an action and the object on which the action is performed. It consists of 244 object classes and 99 action classes with a total of 756 action-object pairwise classes. Following the evaluation of [3] we use the union of the human and object bounding boxes as ground truth for V-HICO.

Model	Backbone	Backbone Pretraining Data	Grounding Training Data	VH	Daly	YC	gYT	Avg.
Models trained for grounding with localization supervision:								
TubeDETR[37]†	ResNet101[7], RoBERTa[17]	ImageNet, Visual Genome, Flickr, COCO	VidSTG	-	-	51.63	-	-
STCAT [9]†	ResNet101[7], RoBERTa[17]	-	VidSTG	-	-	55.90	-	-
VideoGrounding-DINO[33]	Swin-L; BERT[4]	O365, OI, GoldG, Cap4M, COCO, RefC	VidSTG	-	-	57.73	-	-
GLIP[13]‡	Swin-L* [18]	-	FourODs,GoldG,Cap24M	66.05	-	52.84	53.62	-
Models trained for grounding with vision-text pairs only:								
CoMMA[29]‡	CLIP	WIT-400M	HT100M	55.20	61.06	52.65	47.56	54.12
RegionCLIP[39]‡	RN50x4*	WIT-400M	CC3M	57.92	67.12	51.56	52.84	57.36
WWW-CLIP[3]	CLIP	WIT-400M	HT100M	60.71	70.08	57.10	55.49	60.85
WWW-CLIP[3]	CLIP*	WIT-400M	HT100M	62.34	71.35	58.35	56.98	62.26
No training for grounding:								
GEM[2]	CLIP	WIT-400M	-	67.79	78.52	50.08	36.92	58.33
	OpenCLIP	LAION2B	-	68.28	74.05	56.87	32.91	58.03
	ViCLIP	InternVid-FLT-10M	-	65.08	73.75	53.62	51.28	60.93
VideoGEM (ours)	CLIP	WIT-400M	-	**76.90**	**84.53**	52.57	47.46	65.37
	OpenCLIP	LAION2B	-	76.42	80.32	**60.05**	45.33	65.53
	ViCLIP	InternVid-FLT-10M	-	75.75	78.25	55.10	**57.21**	**66.58**

Table 1. **Accuracy of VideoGEM compared to the State-of-the-Art.** VideoGEM includes prompt decomposition, and static and dynamic weights. GEM is applied with the same action prompt as VideoGEM. We compare the accuracy on V-HICO (VH), Daly, YouCook-Interactions (YC), and GroundingYouTube (gYT). Finetuned backbones are marked with *. †results from [33]. ‡results from [3].

DALY (Daily Action Localization in YouTube videos) [35] annotates 510 YouTube videos with ten action classes.

GroundingYouTube [3] is based on the MiningYouTube dataset [11] which contains 250 cooking videos from YouTube. GroundingYouTube provides spatio-temporal annotations including bounding boxes for actions.

YouCook-Interactions [29] is based on the validation set of YouCook2 [41] that contains 457 videos. It provides spatial bounding box annotations for interactions with labels as natural language sentences.

4.2. Implementation Details

We use GEM with $K = 7$ self-self attention layers and $J = 1$ iterations according to GEM [2], using the static weights: $w_s = [0.3, 0.4, 0.5, 0.6, 0.7, 0.9, 0.9, 0.9]$ for the seven self-self attention layers, and its initial self attention input. Note that w_s starts at index $L - K$. We further apply dynamic weights for the last $D = 3$ layers. We use a temperature of $\tau_d = 20$ for dynamic weights according to Equation 9 for all datasets and models. For prompt decomposition we apply the weights $w_{verb} = 0.2, w_{obj} = 0.2, w_{act} = 0.6$ according to Equation 13. We evaluate on ViCLIP, OpenCLIP, and CLIP as backbones in a training-free manner meaning that they are not specifically trained for action localization. For ViCLIP, we sample 7 frames around the labeled frame to get a video input. 4 frames are sampled before, 3 frames are sampled after the labeled frame. Accuracy is used as the main evaluation metric. A prediction is correct if the predicted location is inside the ground-truth bounding box, otherwise it is false. The accuracy is calculated as the proportion of the correct predictions.

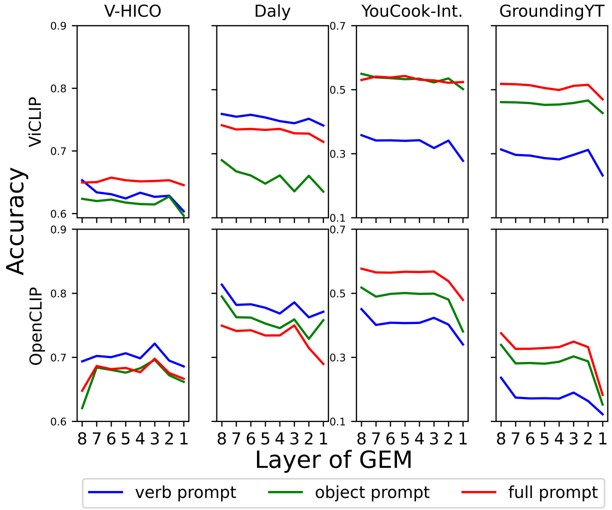

Figure 3. **Importance of GEM layers.** The accuracy of GEM with one removed layer is calculated. The removed layer index is on the x-axis where 1 is the final layer of GEM going down to 8 which is the initial self attention input to GEM.

4.3. Comparison to State-of-the-Art

We first compare the proposed approach to trainable video grounding methods as well as to the vanilla GEM approach in Table 1. To the best of our knowledge, all existing video grounding models are either specifically trained for action

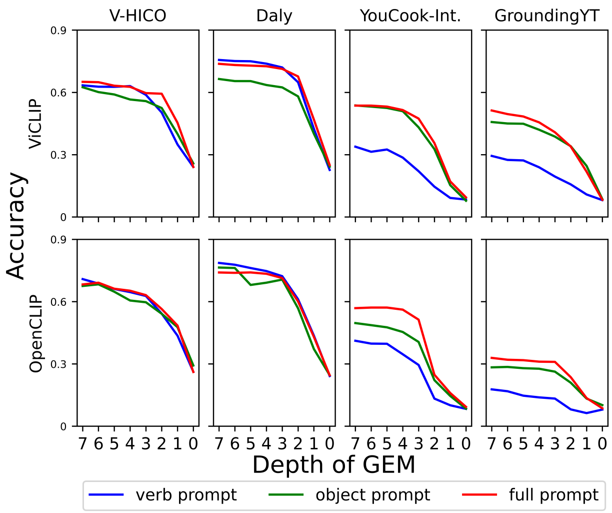

Figure 4. **Influence of the number of GEM layers.** Up to seven layers are added for GEM starting with a self-self attention layer for the final Transformer block. With zero layers, the output equals to the output of the backbone without GEM.

grounding or have a fine-tuned backbone for action localization, whereas our setup is training-free.

First, we show that on average our VideoGEM outperforms all other methods by more than 3% with any of the three backbones. Looking at the different backbones in detail, we notice that especially CLIP and OpenCLIP, so backbones specialized for objects, perform well on the V-HICO and Daly datasets, while ViCLIP significantly outperforms all other backbones on GroundingYouTube. We attribute this to the fact that V-HICO and Daly focus mainly on object annotations, while GroundingYouTube mainly focuses on annotations based on the center point of the action, therefore, stronger deviates from object-centered bounding boxes. Moreover, YouCook-Interactions and GroundingYouTube only contain videos from the specialized cooking domain, while V-HICO and Daly contain more general actions. General actions might be easier recognizable in a zero-shot object-centric setting without specific domain knowledge. Compared to the original GEM pipeline, we show that VideoGEM always improves over GEM independently of the dataset and the backbone that is chosen.

4.4. Ablations

Layer Importance. Figure 3 we first visualize the per-layer impact by excluding single layers from one to eight layer from GEM. We observe that the accuracy is lower when the final layers of GEM are excluded compared to excluding early layers. This might lead to the assumption, that reducing the number of GEM layers would boost performance as well. Compared to that however, Figure 4 shows the results when adding up to seven layers with zero layers corresponding to the output of the backbone without GEM.

Backbone	Weights	VH	Daly	YC	gYT	avg
ViCLIP	none	74.79	76.84	54.38	56.39	65.60
	dyn	74.49	76.85	54.62	56.47	65.61
	stat	**76.18**	**78.38**	55.02	56.75	**66.58**
	s+d	75.75	78.25	**55.10**	**57.21**	**66.58**
OpenCLIP	none	77.86	79.27	59.20	40.17	64.13
	dyn	**78.41**	79.07	59.77	43.29	65.14
	stat	76.12	80.30	**61.82**	42.96	65.30
	s+d	76.42	**80.32**	60.05	**45.33**	**65.53**

Table 2. **Influence of different layer weighting strategies**. VideoGEM with prompt decomposition is applied without layer weighting (none), only static (stat), only dynamic (dyn), or combined weights (s+d) on V-HICO (VH), Daly, YouCook-Interactions (YC), and GroundingYouTube (gYT).

It shows that the performance of GEM depends on the overall number of self-self attention layers and that the accuracy increases for more layers saturating at a depth of seven that we use for all our experiments, also suggested by the original GEM paper [2]. This strengthens the concept of static weights, suggesting that higher layers generally capture complex concepts better than earlier layers, justifying why they should be weighted higher in the final decision. Earlier layers on the other hand can still capture important concepts such that they should not be excluded entirely, but just given less importance for the final decision.

Layer Weighting. To assess the effect of different weighting strategies, we compare static, dynamic, and combined weights (Equation 7, 10, and 12) with using no weights in Table 2 observing a performance increase by about 1% on average for both backbones compared to no layer weighting. We observe that the effect of dynamic weighting depends on the backbone as it relies on representative *[CLS]* tokens for the final layers. If the *[CLS]* token is primarily formed in the last layer, the benefit of dynamic weights is reduced, as seen with the ViCLIP backbone. However, with OpenCLIP, we observe consistent improvements. Notably, on the Grounding YouTube dataset as the most distinct from other object-centric datasets, dynamic weights boost performance by more than 3%. We provide further insights in the supplementary material in Tables 8 to 10.

Prompt Decomposition. We evaluate the effect of prompt decomposition in Table 3 by applying VideoGEM without prompt decomposition for only a verb, object, or action prompt, compared to our proposed prompt decomposition method including all prompts. Averaged over all datasets, using prompt decomposition improves over using only a single (verb, object, or action-) prompt by over 5% independently of the used backbone. Each prompt alone achieves strong performance, supporting the approach of first independently processing prompts and only then aggregating their individual predictions. Our method demonstrates their compatibility for recognizing complex activities effectively.

Backbone	Mode	VH	Daly	YC	gYT	avg
ViCLIP	verb	64.72	76.21	36.54	31.26	52.18
	obj	62.30	68.71	54.94	46.78	58.18
	act	65.68	74.17	52.97	51.99	60.95
	all	**75.75**	**78.25**	**55.10**	**57.21**	**66.58**
OpenCLIP	verb	69.60	80.24	43.45	21.60	53.72
	obj	65.44	77.00	51.21	34.20	56.96
	act	66.77	74.87	57.36	38.38	59.35
	all	**76.42**	**80.32**	**60.05**	**45.33**	**65.53**

Table 3. **Influence of prompt decomposition.** VideoGEM with static and dynamic weights is applied without prompt decomposition to only verb, object (obj), or action (act) prompts. The results are compared to VideoGEM with static and dynamic weights as well as prompt decomposition (dec) on V-HICO (VH), Daly, YouCook-Interactions (YC), and GroundingYouTube (gYT).

Backbone	Mode	VH	Daly	YC	gYT	avg
ViCLIP	none	65.68	74.17	52.97	51.99	60.95
	mul	66.16	76.72	55.91	48.87	61.92
	avg	65.44	75.99	**56.75**	49.96	62.04
	ours	**75.75**	**78.25**	55.10	**57.21**	**66.58**
OpenCLIP	none	66.77	74.87	57.36	38.38	59.35
	mul	68.28	**80.36**	58.40	34.18	60.31
	avg	68.40	78.94	58.04	35.34	60.18
	ours	**76.42**	80.32	**60.05**	**45.33**	**65.53**

Table 4. **Influence of merging strategies for prompt decomposition.** VideoGEM (ours) combines verb, object, and action prompt predictions with a weighted average of the predicted positions. This combination technique is compared to the element-wise multiplication (mul) and element-wise averaging (avg) of heatmaps before taking the highest attention value of the resulting heatmap as prediction. *None* corresponds to the baseline evaluation without prompt decomposition for action prompts with combined weights. We use V-HICO (VH), Daly, YouCook-Interactions (YC), and GroundingYouTube (gYT) for testing.

Additionally, we compare different merging strategies in Table 4 for obtaining the final prediction given the three attention heatmaps for the verb, object, and action-prompt. VideoGEM that averages the predicted positions, is evaluated against averaging or multiplying heatmaps elementwise. The ratio of weights for the verb, object, and action prompt is always $1 : 1 : 3$ independent of the merging strategy. Compared to standard GEM, all merging strategies improve significantly. That suggests that decomposing the prompt into its relevant parts enforces the model to focus on every important part compared to only using the full prompt where the model can neglect the verb and mainly focus on the object [2, 31, 38]. Moreover, averaging positions boosts performance much further compared to merging heatmaps (additively, or multiplicatively). This can be explained by two effects. First, it centers the action by predicting a position between the main parts of the action making it more robust. Second, it has self-correcting abilities. If one prediction is slightly off, the other predictions can drag the wrong prediction back to the others. It can thus be seen as an ensemble model having three votes for the correct prediction instead of just one, improving robustness and accuracy.

Model	Data	VH	Daly	YC	gYT	avg
GEM	vid	65.08	73.75	**53.62**	**51.28**	**60.93**
	img	**65.20**	**74.00**	52.17	48.80	60.04
VideoGEM	vid	**75.75**	78.25	**55.10**	**57.21**	**66.58**
	img	74.19	**78.47**	54.86	55.08	65.65

Table 5. **Influence of video input.** VideoGEM with ViCLIP as backbone is evaluated on image and video inputs. For reference we also evaluate GEM with ViCLIP as a backbone. We evaluate on V-HICO (VH), Daly, YouCook-Interactions (YC), and GroundingYouTube (gYT).

Image vs. Video Data. To determine the importance of video data for action grounding, we compare ViCLIP for video and image input in Table 5. ViCLIP takes eight images as input. In the image setting, we give the same image repeated eight times as an input. For the video setting, we use subsequent frames as input according to Section 4.2. Using video input outperforms its image-based counterpart independently of using standard GEM or VideoGEM by almost 1% on average. While the video input increases accuracy on YouCook-Interactions and GroundingYouTube compared to the image input, it performs similarly on V-HICO and Daly. This can be attributed to the more static actions in V-HICO and Daly like *"snapping fingers"* for V-HICO or *"Drinking"* for Daly, while f.e. *"chopping"* or *"arranging"* in cooking videos as in YouCook-Interactions or GroundingYouTube is more dynamic. Note, that the only difference between the video and image input is, that the image input repeats the same image 8 times, while for video input surrounding frames are used. If the actions are static and the surrounding frames are very similar, the video and image input are also very similar resulting in only minor performance differences.

5. Conclusion

In this work, we introduced VideoGEM, the first training-free method for action grounding in videos. We proposed a weighting technique using static and dynamic weights to assign greater importance to layers that capture conceptually relevant information for complex activities. Additionally, we introduced prompt decomposition to fully leverage action prompts, helping to reduce object bias in standard image- and video-language models. Remarkably, our training-free VideoGEM outperforms all previous state-of-the-art methods that rely on fine-tuning backbones for action localization tasks.

6. Acknowledgments

Walid Bousselham is supported by German Federal Ministry of Education and Research (BMBF) project STCL - 01IS22067. Nina Shvetsova is supported in part by German Federal Ministry of Education and Research (BMBF) project STCL - 01IS22067. Prof. Hilde Kuehne is supported in part by the ERC Starting Grant GraViLa 101117556. We thank Dr. Brian Chen for his support and helpful discussions.

References

[1] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond, 2023.

[2] Walid Bousselham, Felix Petersen, Vittorio Ferrari, and Hilde Kuehne. Grounding everything: Emerging localization properties in vision-language transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3828–3837, 2024.

[3] Brian Chen, Nina Shvetsova, Andrew Rouditchenko, Daniel Kondermann, Samuel Thomas, Shih-Fu Chang, Rogerio Feris, James Glass, and Hilde Kuehne. What when and where? self-supervised spatio-temporal grounding in untrimmed multi-action videos from narrated instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18419–18429, 2024.

[4] Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[5] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[6] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6047–6056, 2018.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[8] Hueihan Jhuang, Juergen Gall, Silvia Zuffi, Cordelia Schmid, and Michael J Black. Towards understanding action recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 3192–3199, 2013.

[9] Yang Jin, Zehuan Yuan, Yadong Mu, et al. Embracing consistency: A one-stage approach for spatio-temporal video grounding. *Advances in Neural Information Processing Systems*, 35:29192–29204, 2022.

[10] Alexander Kläser, Marcin Marszałek, Cordelia Schmid, and Andrew Zisserman. Human focused action localization in video. In *Trends and Topics in Computer Vision: ECCV 2010 Workshops, Heraklion, Crete, Greece, September 10-11, 2010, Revised Selected Papers, Part I 11*, pages 219–233. Springer, 2012.

[11] Hilde Kuehne, Ahsan Iqbal, Alexander Richard, and Juergen Gall. Mining youtube-a dataset for learning fine-grained action concepts from webly supervised video data. *arXiv preprint arXiv:1906.01012*, 2019.

[12] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022.

[13] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10965–10975, 2022.

[14] Shuang Li, Yilun Du, Antonio Torralba, Josef Sivic, and Bryan Russell. Weakly supervised human-object interaction detection in video via contrastive spatiotemporal regions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1845–1855, 2021.

[15] Yi Li, Hualiang Wang, Yiqun Duan, and Xiaomeng Li. Clip surgery for better explainability with enhancement in open-vocabulary tasks. *arXiv preprint arXiv:2304.05653*, 2023.

[16] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.

[17] Yinhan Liu. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 364, 2019.

[18] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[19] Pascal Mettes and Cees GM Snoek. Spatial-aware object embeddings for zero-shot localization and classification of actions. In *Proceedings of the IEEE international conference on computer vision*, pages 4443–4452, 2017.

[20] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2630–2640, 2019.

[21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[22] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training

next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022.

[23] Ling Shao, Simon Jones, and Xuelong Li. Efficient search and localization of human actions in video databases. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(3):504–512, 2013.

[24] Jing Shi, Jia Xu, Boqing Gong, and Chenliang Xu. Not all frames are equal: Weakly-supervised video grounding with contextual similarity and visual clustering losses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10444–10452, 2019.

[25] Nina Shvetsova, Anna Kukleva, Xudong Hong, Christian Rupprecht, Bernt Schiele, and Hilde Kuehne. Howtocaption: Prompting llms to transform video annotations at scale. In *European Conference on Computer Vision*, pages 1–18. Springer, 2025.

[26] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012.

[27] Khurram Soomro, Haroon Idrees, and Mubarak Shah. Action localization in videos through context walk. In *Proceedings of the IEEE international conference on computer vision*, pages 3280–3288, 2015.

[28] Khurram Soomro, Haroon Idrees, and Mubarak Shah. Online localization and prediction of actions and interactions. *IEEE transactions on pattern analysis and machine intelligence*, 41(2):459–472, 2018.

[29] Reuben Tan, Bryan Plummer, Kate Saenko, Hailin Jin, and Bryan Russell. Look at what i'm doing: Self-supervised spatial grounding of narrations in instructional videos. *Advances in Neural Information Processing Systems*, 34:14476–14487, 2021.

[30] Tuan Hue Thi, Jian Zhang, Li Cheng, Li Wang, and Shinichi Satoh. Human action recognition and localization in video using structured learning of local space-time features. In *2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 204–211. IEEE, 2010.

[31] Felix Vogel, Nina Shvetsova, Leonid Karlinsky, and Hilde Kuehne. Vl-taboo: An analysis of attribute-based zero-shot capabilities of vision-language models. *arXiv preprint arXiv:2209.06103*, 2022.

[32] Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, et al. Internvid: A large-scale video-text dataset for multimodal understanding and generation. *arXiv preprint arXiv:2307.06942*, 2023.

[33] Syed Talal Wasim, Muzammal Naseer, Salman Khan, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Videogrounding-dino: Towards open-vocabulary spatio-temporal video grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18909–18918, 2024.

[34] Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Learning to track for spatio-temporal action localization. In *Proceedings of the IEEE international conference on computer vision*, pages 3164–3172, 2015.

[35] Philippe Weinzaepfel, Xavier Martin, and Cordelia Schmid. Human action localization with sparse spatial supervision. *arXiv preprint arXiv:1605.05197*, 2016.

[36] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. *arXiv preprint arXiv:2109.14084*, 2021.

[37] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Tubedetr: Spatio-temporal video grounding with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16442–16453, 2022.

[38] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? 2023.

[39] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, et al. Regionclip: Region-based language-image pretraining. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16793–16803, 2022.

[40] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *European Conference on Computer Vision*, pages 696–712. Springer, 2022.

[41] Luowei Zhou, Chenliang Xu, and Jason Corso. Towards automatic learning of procedures from web instructional videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.